Project Proposal Form
Institute of Information Technology (IIT)
University of Dhaka

Student's Name:	**Amit Kumar Roy**		
Student's Roll:	**1314**	**Phone:**	**01732277419**

Project Title: PyTestGenie: A tool for Automated Unit Test Generation, Smell Detection & Refactoring for Python

Project Description

Testing plays a vital role in ensuring the reliability and maintainability of software applications. But writing unit tests manually is time-consuming, and verifying their correctness and quality is equally challenging. Since Python is a dynamically typed language, automatic unit test generation poses additional difficulties compared to statically typed languages such as Java. This project aims to develop a pipeline-based tool for python that allows developers or researchers to generate unit tests, detect test smell and refactor.

Problem Statement and Objectives

Current Python test frameworks and smell detection tools face these challenges like many are designed for specific IDEs, few tools provide automated refactoring for detected test smells, collaborative & agent-based LLM workflows are not leveraged for improved detection and correction.

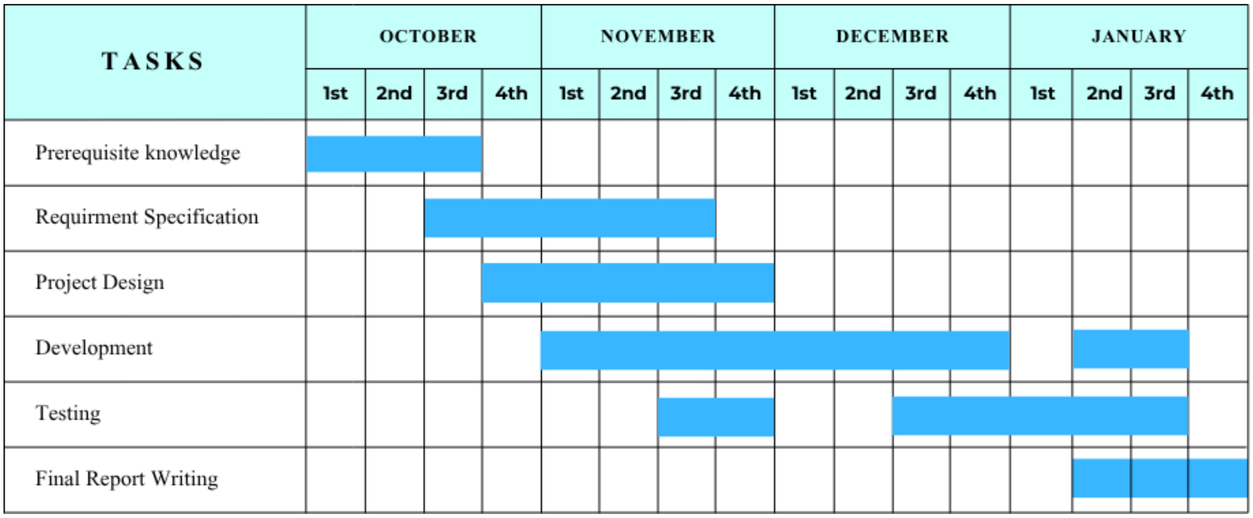
The objectives of PyTestGenie are to generate unit tests[1], develop rule based[3] and LLM-powered agentic framework capable of detecting test smells[2] and automatically refactoring[2] them to improve code quality and maintainability. The framework aims to support multi-agent (1-4 agents) workflows[2], where multiple model agents collaborate to enhance detection accuracy and overall coverage.

Proposed Solution

PyTestGenie is a pipeline framework with the following components:

- Code Input Module: Users can upload files or paste code.
- Test Generation Engine: Users can use both rule-based and LLM-based methods for test generation of the provided code[1].
- Agentic LLM or rule based Test Smell Detection: Detects test smells (such as conditional test logic, duplicate assert etc.) using rule based method[3] and LLM[2]. And also builds multi-agent workflows (1–4 agents) to improve test accuracy and coverage[2].
- Automated Refactoring Module: Refactors detected test smells automatically, producing clean, maintainable tests[2].

Timeline:



TASKS	OCTOBER				NOVEMBER				DECEMBER				JANUARY			
	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
Prerequisite knowledge	🟦	🟦	🟦													
Requirment Specification			🟦	🟦	🟦	🟦	🟦									
Project Design				🟦	🟦	🟦	🟦	🟦								
Development					🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦		🟦	🟦	
Testing							🟦	🟦			🟦	🟦	🟦	🟦	🟦	
Final Report Writing														🟦	🟦	🟦

References:

[1] S. Lukasczyk and G. Fraser, “Pynguin: Automated unit test generation for python,” in Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings, ser. ICSE ’22, Pittsburgh, Pennsylvania: Association for Computing Machinery, 2022, pp. 168–172, isbn: 9781450392235. doi: 10.1145/3510454.3516829. [Online]. Available: https: //doi.org/10.1145/3510454.3516829

[2] R Melo, P Simões, R Gheyi, M d'Amorim, M Ribeiro, G Soares, E Almeida, E Soares, “Agentic LMs: Hunting Down Test Smells”arXiv preprint arXiv:2504.07277, 2025.

[3] D Fernandes, I Machado, R Maciel, “TEMPY: Test smell detector for Python”. Proceedings of the XXXVI Brazilian Symposium on Software Engineering, 2022

Languages or Tools to be used: Python, FastAPI, React.js, GitHub

Supervisor’s Name: Toukir Ahmed

Signature of the supervisor: ____________________

Date: 31 October, 2025

Proposal Presentation Feedback:

Midterm Presentation Feedback: